Salience-Based Adversarial Attacks for Empirical Evaluation of NLP Classification Robustness

Stanford CS224N Custom Project

Fletcher Newell
Department of Computer Science
Stanford University
flnewell@stanford.edu

Abstract

As natural language processing (NLP) models gain widespread use, it is increasingly important to verify consistent model outputs, but due to the discrete nature of textual inputs and large sizes of these models, formal verification methods are intractable. This paper aims to address these limitations by proposing a strategy that leverages a test suite of salience-based adversarial attacks to provide empirical evaluation of robustness properties. We assess our test suite on one single-label model and two multi-label models, and comparison against random and genetic alteration baselines show that our approach excels in assessing robustness of synonym word-swaps and paraphrase-swaps while remaining competitive for evaluating robustness of spelling word-swaps.

1 Key Information to include

My assigned mentor is Tony Wang. All work for this project is my own and was done exclusively for CS224N.

2 Introduction

To trust that a model behaves reasonably, it is important for the model to not only be correct with high probability but to also be consistent with itself, a property known as robustness. Formally, a model is said to be robust if $||y - y'||_1 \leq \delta$ for all y and for all y', where y is the ground truth score of some input, y' is the score of a similar input, and δ is some small, predefined value (Rauber et al., 2018), meaning a model is robust if it gives similar outputs for similar inputs. The notion of robustness is especially important in situations where a model is used to make life-impacting decisions, and the field of artificial intelligence verification has spent considerable resources developing formal methods for proving robustness of general models. Over the last few years, natural language processing (NLP) models have gained widespread use in multiple domains, where the outputs of these models are now used to make decisions that have direct impacts on people, but proving the robustness of natural language processing (NLP) models presents several new challenges.

Three of the greatest challenges to robustness verification of NLP models are the discrete input spaces, unclear notions of similarity, and large model sizes. Formal robustness verification for non-language models are able to perform optimization-like searches on the model inputs to find counterexamples, with the absence of counterexamples providing a mathematical guarantee of robustness. Unlike most other domains, NLP models have discrete inputs in the form of words, meaning that searching for a counterexample requires changing a word in its entirety rather than increasing or decreasing some numerical value (Ren et al., 2019). Furthermore, it is not entirely clear what is considered a similar input for an NLP model. Most non-language models can measure input similarity by subtracting the values of two inputs, but subtracting two sentences has no mathematical meaning. Lastly, modern verification methods are still limited to models of a couple thousand neurons whereas NLP models frequently approach millions of neurons, making these traditional methods intractable (Goyal et al., 2023).

3 Related Work

In the absence of tractable formal methods, evaluation efforts have turned to empirical methods that can provide assurance of robustness properties. Since checking over all alterations of an input is similarly intractable, adversarial generation is a popular method for choosing alterations to an input that are most like to violate the robustness property while remaining similar to the original input. Prior work has identified two notions of similarity for text-based inputs: inputs that have similar literacy meanings and inputs that visually look similar (Zhang et al., 2020). For the first notion, adversarial attacks choose alterations that preserve the meaning of a sentence, swapping words for their synonyms or interchanging clauses that convey the same ideas in different words. In the second notion, adversarial attacks choose alterations that remove, insert, or swap the characters in a word that a human would be unlikely to notice. Although some work allows these character alterations to result in non-words (Morris et al., 2020b), we will require that a valid alteration be a word since simple spellchecking software should identify any non-words.

There are many strategies for adversarial generation, all of which attempt to shift model predictions while remaining undetectable to humans. Of particular interest to this paper are classification attacks, where the target model's output is some numerical prediction or score. One of the simplest attack methods is to choose random sections of an input and make the best possible alterations to those sections as determined by running the altered texts through the model, but this approach is highly non-deterministic and does not make adequate use of available information (Roth et al., 2021). For cases in which the model's entire structure is visible, cases in which we call the model a white-box, gradient-based methods leverage the weights and layers of the model to identify the best alterations to make to an input (Goyal et al., 2023); however, this approach can be slow and is still outperformed by other methods, like genetic algorithms and particle swarm optimization. Both the genetic and particle swarm approaches make use of traditional discrete optimization algorithms to maintain a population — in this case a selection of alterations — and use model queries to maximize the difference in output prediction scores between the altered and unaltered inputs (Qiu et al., 2022). These methods require no information of the model's structure beyond assuming that the model can be queried and that the prediction score of each label is visible after the query, so these methods are applicable even when the model structure is not visible, a case in which we call the model a black-box.

Of particular interest to this paper is a more-recent method known as Probability Weighted Word Saliency, sometimes shortened to PWWS. Salience-mapping is a popular method of model explainability, allowing analysts to interpret the decision-making process of text models by evaluating the impact of each word in a given input on the corresponding output (Li et al., 2016). Work by Ren et al. (2019) leveraged this salience-mapping as a heuristic for choosing which words to replace with their synonyms, greedily replacing the most impactful words with the synonym that maximizes output score difference, as found through queries to the model. The key assumption here is that an unimportant word in the original input will remain unimportant even when replaced with its synonyms, so the most impactful words are targeted as the most vulnerable to attacks. This salience-based method reportedly outperformed gradient, genetic, and particle swarm approaches, but the work limits its evaluation to synonym word-swapping in single-label classification tasks. While synonym word-swapping is an important test for robustness, we also want to evaluate a model's robustness with respect to other input alterations, such as misspellings and clause-level differences. We therefore seek to expand this method to other forms of attacks while also generalizing to multi-label models.

4 Approach

In this paper, we aim to assess the effectiveness of salience-based adversarial generation in empirical robustness evaluation by creating a test suite of attacks that extend the single-label synonym word-swapping salience approach created by Ren et al. (2019) to multi-label classification models. We consider three types of alterations for our test suite:

1. **Synonym Word-Swapping**: To simulate various user vocabularies, some words from a given input are swapped with their closest synonyms, as defined by the Natural Language Toolkit (Bird et al., 2009).

2. **Spelling Word-Swapping**: To simulate misspellings that would be uncaught by spellchecking software, some words from a given input are swapped with the closest-spelled words, as defined by the Levenshtein Distance. To ensure the alterations look visually similar, we limit available word swaps to words that share the same first and last letter and have a Levenshtein Distance of one, meaning that one letter inside the word is replaced, inserted, or deleted.
3. **Paraphrasing**: To simulate differences in user writing styles, we replace entire clauses with other variations of that clause. We use the Stanza library to identify and extract the clauses from the text (Qi et al., 2020), and the alterations of clauses are then generated by passing the identified clauses into the Parrot Paraphraser model (Damodaran, 2021).

We implement our own method for salience-based multi-label adversarial generation, extending the logic of the single-label approach outlined by Ren et al. (2019). For now, consider the case of word-swaps. We first developed a salience interpreter from scratch that determines the importance of each word in an input text by replacing that word with an out-of-vocabulary mask, running the model on this altered text, and comparing the resultant prediction scores to that of the original text. Formally, let x represent our n-length input text such that $x = (w_1, w_2, \dots, w_n)$ where w_i is the ith word in x, and let $\hat{x}_j = (w_1, \dots, w_{j-1}, [MASK], w_{j+1}, \dots w_n)$. For any label y, we define the salience of the jth word in x as

$$sal(j, x, y) = \Pr[y|x] - \Pr[y|\hat{x}_j] \tag{1}$$

where $\Pr$ is the prediction score function of the target model.
We use our salience interpreter to identify the most impactful word of the input,

$$\text{argmax}_{j\in[1,n]} \sum_{y\in Y} |sal(j, x, y)| \tag{2}$$

where Y is the set of all labels. We then greedily select the alteration — either the synonym or close spelling, depending on the attack type — of the jth word that maximizes the difference in output prediction scores. That is, for a set of alterations A of the jth word, we select

$$\text{argmax}_{a\in A} \sum_{y\in Y} |\Pr[y|x] - \Pr[y|x_{j=a}]| \tag{3}$$

where $x_{j=a}$ represents the string x with the jth word replaced by alteration a. We then fix this alteration and repeat the process until the desired number of alterations has been met. To better preserve the original meaning of the text, we exclude previously-altered words when determining which word in the text to alter next. The process is the same for the paraphrasing attack except that we mask the entire clause to determine its salience and then greedily select the paraphrase of the most salient clause that maximizes the difference in model predictions.

Although there exist many popular libraries for generating adversarial attacks against language models, these libraries work exclusively for single-label classification since most attack methods, such as popular gradient-based methods, are not well conditioned for cases where there can be multiple correct labels (Morris et al., 2020b). For this reason, we also implement the following two baselines ourselves:

1. **Random**: Based on the algorithm described by Roth et al. (2021), the index of a word or clause to be swapped is randomly generated from among all valid indices (that is, indices for which there is at least one possible swap), and all alterations to that index are run through the model, greedily selecting the alteration that causes the greatest difference in the model's output prediction scores. This is mostly the same as our salience-based method except that we use no heuristic, instead choosing the words and clauses at random.

2. **Genetic**: Possible alterations to an input are treated as variables in an optimization problem solved with a discrete genetic algorithm, where the function to be maximized is the difference in model prediction scores on the altered and unaltered texts. While the method is our own variation of existing work for single-label cases (Morris et al., 2020a), we use the Python GeneticAlgorithm library as our optimizer (Solgi, 2020).

5 Experiments and Analysis

We apply our attacks to three pretrained models, allowing us to test our methods on real-world near-state-of-the-art models while maintaining the traditional verification environment in which the analysts are not the developers. We test on one single-label model and two multi-label model to prove the flexibility of our methods.

- **Roberta Spam** (Shenoda, 2023a): A single-label Roberta model trained on Shenoda's Spam Messages dataset (Shenoda, 2023b). Given a message, the model predicts whether or not the message is spam.
- **Tweet Topic 21 Multi** (Antypas et al., 2022b): A multi-label TimeLMs model trained on the TweetTopic dataset (Antypas et al., 2022a). Given a Tweet, the model predicts the topics of the tweet from 21 topic labels.
- **Roberta-Base Go Emotions** (Lowe, 2023): A multi-label Roberta-Base model trained on the Go Emotions dataset (Demszky et al., 2020). Given a social media post or comment, the model predicts the emotions of the user from 22 emotion labels.

For each model, we test our methods on 300 randomly-selected inputs, each of at least 30 words, drawn from the testing data of the corresponding dataset. For both the synonym word-swapping and spelling word-swapping attacks, we allow up to three words to be altered in each input. In the paraphrasing attack, we allow only two clauses to be swapped for their paraphrases. Since each input is at least 30 words, these limits ensure that the altered input is still closely similar to the original input.

We consider two metrics for measuring method performance: average score distance and maximum score distance. Average score distance is the mean distance between prediction scores of the original and altered texts across all 300 inputs, giving insight into the consistency of a method. Maximum score distance is the greatest distance between prediction scores of the original and altered texts from all 300 inputs, showing the worst-case model performance that the method was able to achieve. Note that for both metrics, a higher value denotes a better attacking method and lower confidence of model robustness.

In practice, both metrics are useful for evaluating robustness. Maximum score distance can be a definitive counterexample if greater than the defined tolerance threshold and is therefore in itself sufficient for proving the lack of model robustness. In non-critical settings, however, we may wonder whether the maximum score distance is an outlier and therefore value the average score distance as a means to understand how well the typical attack is expected to perform.

We present the results of our experiments in the tables below. For easy interpretability, we convert the raw distances of the attack scores to percentages over the probability domains, such that every score is between 0 and 1 with a higher value still indicating a better attack.

Synonym Word-Swaps: Average Score Distance			
	Random (Baseline)	Genetic (Baseline)	Salience
Roberta Spam	0.0077	0.0136	**0.0233**
Tweet Topic 21 Multi	0.0296	0.0878	**0.1046**
Robert Base Go Emotions	0.0619	0.1015	**0.1299**

Table 1: Average score distances for synonym word-swaps across 300 trials.

Synonym Word-Swaps: Maximum Score Distance			
	Random (Baseline)	Genetic (Baseline)	Salience
Roberta Spam	0.0487	0.0426	**0.0703**
Tweet Topic 21 Multi	0.1684	0.1712	**0.2595**
Robert Base Go Emotions	0.2092	0.2736	**0.3951**

Table 2: Maximum score distances for synonym word-swaps across 300 trials.

Tables 1 and 2 show that salience-mapping adversarial attacks consistently outperform both baselines for average attack efficiency (average score distance) and best attack efficiency (maximum score distance), making it the better method for testing robustness relative to synonym word-swapping.

Spelling Word-Swaps: Average Score Distance			
	Random (Baseline)	Genetic (Baseline)	Salience
Roberta Spam	0.0482	**0.0934**	0.0706
Tweet Topic 21 Multi	0.0787	**0.1158**	0.1081
Robert Base Go Emotions	0.0959	**0.1883**	0.1724

Table 3: Average score distances for spelling word-swaps across 300 trials.

Spelling Word-Swaps: Maximum Score Distance			
	Random (Baseline)	Genetic (Baseline)	Salience
Roberta Spam	0.1143	**0.2223**	0.2008
Tweet Topic 21 Multi	0.2106	**0.3832**	0.3678
Robert Base Go Emotions	0.3490	0.5408	**0.6072**

Table 4: Maximum score distances for spelling word-swaps across 300 trials.

The results of the spelling word-swapping attacks are shown in Table 3 and Table 4. We see from these results that the salience-based method is generally outperformed by the genetic baseline, but this is not surprising considering the disruptive nature of spelling-level alterations. In synonym word-swapping, the available alterations should preserve the meaning of a word, so we expect that the synonyms of an unimportant word will also be unimportant. Whenever we consider spellings, however, we can no longer apply this same logic since an inconsequential word may be just a single character away from a word that changes the meaning of the sentence entirely. While the salience-based method is liable to miss these cases since it selects alterations greedily based on original word importance, the genetic algorithm attack gives no preference to original word importance and is therefore able to find and exploit alterations in which a word switches from inconsequential to important. Even despite this difference, however, we note that the salience-based method reliably outperforms the random baseline and still remains competitive with the genetic baseline.

Paraphrasing: Average Score Distance			
	Random (Baseline)	Genetic (Baseline)	Salience
Roberta Spam	0.0165	0.0197	**0.0280**
Tweet Topic 21 Multi	0.0471	0.0984	**0.1108**
Robert Base Go Emotions	0.0866	0.1185	**0.1541**

Table 5: Average score distances for paraphrasing across 300 trials.

Paraphrasing: Maximum Score Distance			
	Random (Baseline)	Genetic (Baseline)	Salience
Roberta Spam	0.0424	0.0773	**0.1038**
Tweet Topic 21 Multi	0.1831	0.2264	**0.2706**
Robert Base Go Emotions	0.2162	0.2510	**0.3422**

Table 6: Maximum score distances for paraphrasing across 300 trials.

We see from Tables 5 and 6 that the salience-based method extends well to paraphrasing attacks, consistently outperforming both baselines for average score distance and maximum score distance. We suspect that this is because paraphrasing, like synonym word-swapping, preserves importance such that paraphrases of an inconsequential clause will also be inconsequential, allowing our method to focus on the most impactful clauses and optimize greedily with respect to clause salience.

Overall, we find that the salience-based attack works well for multi-label classifiers. Although the method even performed better on the multi-label models (Tweet Topic 21 Multi and Robert Base Go Emotions) than on the single-label model (Roberta Spam), we speculate that the Roberta Spam model was simply more difficult to attack — and consequentially more robust — since both baselines also performed worse on Roberta Spam than the other two models. We additionally find that the salience-based method works well on both synonym word-swapping and paraphrasing attacks, outperforming both baselines. This high performance makes our salience-based method ideal for evaluating NLP models for robustness with respect to semantic meaning.

In a practical setting, we would use the results of our attacks to make statements regarding the robustness of each model. The Roberta Spam model was the most robust of the three models, as evidenced by consistently having the lowest score distances. Conversely, the Robert Base Go Emotions model was the least robust, with the semantic-preserving attacks of synonym word-swapping and paraphrasing capable of shifting the total prediction scores by over 30 percent. The model was even less robust with respect to similar visual appearances, with some spelling word-swap attacks causing the total prediction scores to shift by over 50 percent. Tweet Topic 21 Multi was moderately robust, resisting all three attack types better than Roberta Base Go Emotions but still more vulnerable to changes relative to Roberta Spam. We note that all three models were least robust to spelling word-swaps, but this is expected since similar-looking words can have vastly different meanings and therefore cause larger disturbances in model predictions. Nevertheless, this vulnerability to misspellings suggests that NLP applications may wish to consider adding processes that can better identify and guard against words that may not make sense in the context in which they appear.

6 Conclusion

In this paper, we have presented a test suite of salience-based adversarial generation methods that leverage word importance to design attacks against NLP models. Our salience-based approach outperforms random and genetic baselines on synonym word-swapping and paraphrasing attacks and remains competitive with the genetic baseline on spelling word-swapping while outperforming the random baseline method.

Notably, our work makes no assumption regarding the structure of the NLP model beyond requiring that the prediction score of each label is visible, making the work applicable to even situations in which the model weights are inaccessible. The foremost limitation of this work is that these methods provide no mathematical guarantee of robustness in the absence of successful attacks. A large number of unsuccessful attacks certainly build confidence in the model's robustness, but it could simply be the case that the texts we choose to alter are inherently difficult to attack whereas other texts might be easier to manipulate. Instead of randomly sampling the testing data for input texts as we did in this paper, we therefore recommend that future work investigate how these input texts might be strategically chosen. This could not only increase attack effectiveness but also provide some guarantee of robustness in the absence of successful attacks.

We also acknowledge that our synonym word-swapping method is subject to the same critique as most other work on the subject; namely, there are cases where swapping a word for a synonym would be obvious to a human. Such is the case of "elementary school", where a human would be quick to note the awkwardness of "uncomplicated school" despite "elementary" and "uncomplicated" being synonyms. Future work might consider replacing the NLTK synonym dictionary with a dictionary that better encompasses these multi-word concepts.

Lastly, we note that the use of a generative text model to create paraphrases for clause-level attacks injects a reliance on the generative model that may be unacceptable in some domains due to concerns of hallucination. We therefore recommend that future work investigate deterministic methods for creating reasonable paraphrases.

Our overall contributions to the field include the development of salience-based attacks for multi-label classifiers, the benchmarking of these attacks on three real-world models, and the assessment of their usefulness in evaluating model robustness with respect to synonym word-swapping, spelling word-swapping, and paraphrasing. Through this work, we hope that verification analysts will one day be able to provide robustness guarantees for the language models that are gaining popularity across the world.

References

Dimosthenis Antypas, Asahi Ushio, Jose Camacho-Collados, Leonardo Neves, Vitor Silva, and Francesco Barbieri. 2022a. Twitter Topic Classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Dimosthenis Antypas, Asahi Ushio, Jose Camacho-Collados, Vitor Silva, Leonardo Neves, and Francesco Barbieri. 2022b. Twitter topic classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3386–3400, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Prithiviraj Damodaran. 2021. Parrot: Paraphrase generation for nlu.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *58th Annual Meeting of the Association for Computational Linguistics (ACL).*

Shreya Goyal, Sumanth Doddapaneni, Mitesh M Khapra, and Balaraman Ravindran. 2023. A survey of adversarial defenses and robustness in nlp. *ACM Computing Surveys*, 55(14s):1–39.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp.

Sam Lowe. 2023. Roberta-base go emotions.

John X Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020a. Reevaluating adversarial examples in natural language. *arXiv preprint arXiv:2004.14174.*

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020b. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages.

Shilin Qiu, Qihe Liu, Shijie Zhou, and Wen Huang. 2022. Adversarial attack and defense technologies in natural language processing: A survey. *Neurocomputing*, 492:278–307.

Jonas Rauber, Wieland Brendel, and Matthias Bethge. 2018. Foolbox: A python toolbox to benchmark the robustness of machine learning models.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Tom Roth, Yansong Gao, Alsharif Abuadbba, Surya Nepal, and Wei Liu. 2021. Token-modification adversarial attacks for natural language processing: A survey. *arXiv preprint arXiv:2103.00676.*

Michael Shenoda. 2023a. Roberta based spam message detection.

Michael Shenoda. 2023b. Spam messages dataset.

Ryan Solgi. 2020. geneticalgorithm.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.